

# Universitat Politècnica de Catalunya

Escola Superior d'Enginyeria Industrial,  
Aeroespacial i Audiovisual de Terrassa



Treball Final de Grau

Implementació d'un Site Web

Titulació: Grau en Enginyeria de Sistemes Audiovisuals  
Autor: Pablo López Aladid  
Director : Jorge Martín  
10/6/2019

# Índex

<i>Taula de figures</i>	4
<i>Declaració d'honor</i>	5
<i>Introducció</i>	6
<i>Objectius</i>	6
<i>Abast</i>	6
<i>Requeriments</i>	7
<i>Justificació del Treball</i>	7
<i>Diagrama de Gantt</i>	8
<i>Estat de l'art</i>	9
- <i>Què és un servei de hosting?</i>	9
- <i>Què és una Raspberry Pi ?</i>	9
- <i>Què és un controlador Arduino ?</i>	10
<i>Configuracions</i>	11
- <i>Configuració de la Raspberry Pi</i>	11
- <i>Configuració del Servidor</i>	12
- <i>Configuració del Router</i>	14
<i>Site Web</i>	15
<i>Programació de la placa Arduino</i>	23
<i>Creació de la base de dades</i>	28
<i>Circuit de l'estació climatològica</i>	29

<i>Seguretat implementada al Site</i>	31
<i>Connexió a la base de dades</i>	32
<i>Pressupost</i>	35
<i>Conclusions i accions futures</i>	36
<i>Codis</i>	37
- <i>Codi primera pàgina del site</i>	37
- <i>Codi segona pàgina del site</i>	43
- <i>Codi Arduino</i>	46
- <i>Codi css del site</i>	50
<i>Bibliografia</i>	53

## Taula de figures

<i>Figura 1: Diagrama de Gannt</i>	8
<i>Figura 2: Ports del router</i>	14
<i>Figura 3: Pàgina de Login</i>	15
<i>Figura 4: Primera pàgina del site</i>	17
<i>Figura 5: Primera pàgina del site</i>	17
<i>Figura 6: Resultats del cercador</i>	18
<i>Figura 7: Segona pàgina del site</i>	22
<i>Figura 8: Sensor DHT11</i>	23
<i>Figura 9: Mòdul TinyRTC</i>	24
<i>Figura 10: Esquema de connexions</i>	29
<i>Figura 11: Muntatge</i>	30
<i>Figura 12: Càlculs del pressupost</i>	35

## Introducció :

La idea d'aquest projecte és crear una web a un servidor propi. Aquesta web ens servirà per emmagatzemar dades climatològiques d'una localització (temperatura i humitat, juntament amb les dades temporals de la mesura) preses prèviament amb una placa Arduino i el circuit corresponent. Això ens permetrà saber en tot moment les condicions de temperatura i humitat de la localització escollida, fins i tot estant fora. La finalitat és crear un entorn previ per aplicacions de domòtica.

## Objectius :

Podem, doncs, establir un seguit d'objectius per assolir durant els mesos en que treballarem en aquest projecte. Es dividiran en tres : instal·lació del servidor, muntatge del circuit i programació. El primer es tracta de posar en marxa un servidor propi, el segon de dissenyar el circuit adequat amb la placa i els sensors escollits i finalment l'apartat de programació consisteix en dissenyar la web i el codi per l'Arduino.

## Abast:

- Recerca d'informació sobre els dispositius a utilitzar
- Adquisició dels dispositius
- Implementació del servidor
- Connexió del servidor en remot
- Programació del site
- Disseny i programació de la placa Arduino
- Documentació

## Requeriments:

Aquest projecte es durà a terme tenint en ment dues idees centrals. D'una banda volem fer una inversió el més petita possible, i d'altra volem crear un entorn capaç de recollir dades i pujar-les a un servidor per poder consultar-les des d'una localització remota. A més, es volen utilitzar tecnologies de codi obert amb suport de la informació lliure que trobem a internet, de forma que no es necessiti ajuda de cap entitat privada.

Es necessiten equips amb capacitat de connexió a internet mitjançant xarxes WiFi, els sensors i mòduls adequats pel disseny del circuit i el cablejat per fer la instal·lació a la protoboard.

## Justificació del Treball:

L'objectiu bàsic que cobreix aquest projecte és de l'àmbit de la domòtica. Crearem l'entorn per portar un control de la temperatura i humitat dins d'un habitatge amb la possibilitat de consultar les dades des de fora. Els avantatges són econòmics, ja que el sistema s'implementa a un servidor propi amb una Raspberry Pi i amb un Arduino per fer les mesures. S'aprofitarà el coneixement obtingut a internet, atès que són tecnologies molt utilitzades amb una gran comunitat d'investigadors al darrere.

## Diagrama de Gantt:

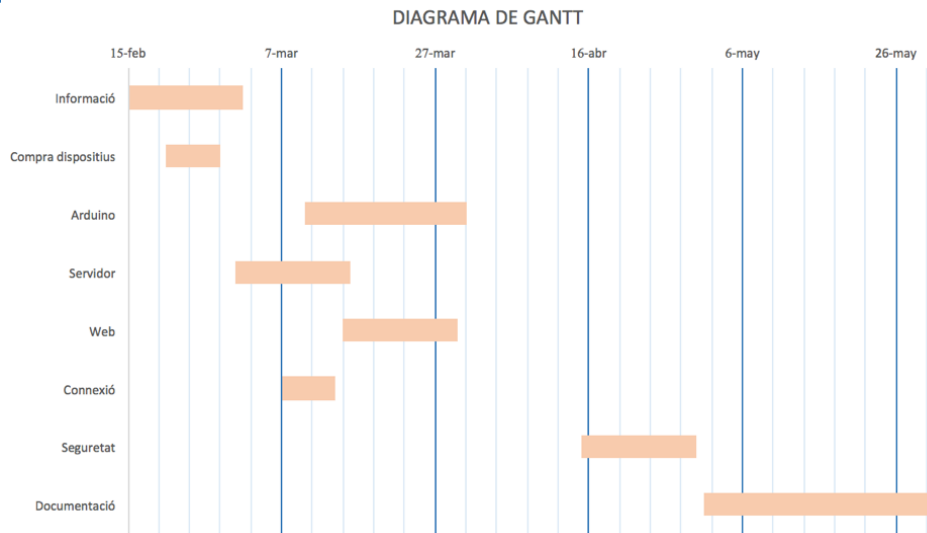


Figura 1:Diagrama de Gantt

S'ha estructurat el temps del projecte de forma que el primer que fem és buscar informació sobre quins dispositius comprar i, un cop adquirits, comencem a treballar. Primerament programarem el servidor i la seva connexió, i després de forma pràcticament paral·lela dissenyem el circuit de la placa Arduino i el site web. Més tard implementem la seguretat a la web i finalment realitzem la documentació.

## Estat de l'art

Una de les opcions per implementar el servidor és utilitzar un ordinador normal. Això ens donaria més potència, però no la necessitem i la inversió pujaria notablement. També podríem contractar un servei de hosting, però a la llarga la inversió pujaria.

La solució més viable, doncs, és usar un mini ordinador. Escollim la Raspberry Pi per motius econòmics i de suport per part dels creadors. Com hem comentat prèviament l'ús de plaques Arduino i Raspberry Pi està molt estès. Això és degut al seu cost realment econòmic i a la gran comunitat d'investigadors i desenvolupadors que s'ha creat al voltant. Pel que fa el disseny de l'estació meteorològica s'aprofitaran diverses idees extretes d'internet, ja que hi podem trobar molts exemples realment funcionals.

De cara a muntar la Raspberry com a servidor també s'utilitzaran coneixements trobats a la xarxa, atès que és un dels usos més habituals d'aquests dispositius.

### Què és un servei de hosting?

L'allotjament web és un servei que proveeix els usuaris d'un sistema per poder emmagatzemar informació a un ordinador que es troba a la xarxa. Ens permet tenir una web a canvi d'una mensualitat. És una bona opció, atès que simplifica el procés. El problema es troba en que és impossible garantir un funcionament correcte el 100% del temps.

### Què és una Raspberry Pi ?

Una Raspberry Pi és un ordinador SBC. Aquestes sigles provenen de l'Anglès i signifiquen *Single Board Computer*, és a dir, és un ordinador monoplaca. És molt utilitzada amb finalitats didàctiques en l'àmbit de la informàtica degut al seu preu reduït i les seves interessants característiques, com la potència de càlcul o el suport per part de la pròpia fundació Raspberry Pi pel que fa la instal·lació de distribucions de Linux. Podem trobar diferents models de Raspberry Pi, sent la Raspberry Pi 3 model B+ la que s'ha utilitzat en aquest projecte.



S'ha escollit aquest model degut a que és l'últim en el moment d'escriure aquest treball, per tant ens assegura que encara tindrà vida útil un cop s'acabi i l'ordinador podrà ser reutilitzat en un futur. Les característiques d'aquest model són :

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth
- 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pins
- HDMI
- 4 ports USB 2.0
- Port CSI y DSI per connectar una càmera y una pantalla tàctil
- Sortida d'àudio estèreo y vídeo compost.
- Micro-SD
- Power-over-Ethernet (PoE)

### Què és un controlador Arduino ?

Un Arduino és, en resum, una placa de circuit imprès amb un microcontrolador de codi obert. Aquesta placa incorpora un conjunt de processadors variats i un seguit de pins digitals i analògics d'entrada i sortida. La programació d'aquests controladors es realitza mitjançant un entorn de desenvolupament propi d'Arduino amb un codi semblant a C++.

S'escull aquest tipus de controlador degut a la gran comunitat de programadors que hi ha al darrere, ja que ens facilita molt la feina el fet de tenir llibreries dedicades i fòrums on es resolen dubtes.

Igual que amb les Raspberry Pi, i de manera encara més extensa, les plaques Arduino tenen diferents models. Per fer aquest projecte s'ha escollit una placa no oficial per motius econòmics i tècnics. El motiu de més pes és que la placa escollida porta integrat un mòdul WI-FI, necessari per poder enviar les dades al servidor.

La placa escollida és la *NodeMCU esp8266*. Les seves característiques són :

- 32-bit RISC CPU: Tensilica Xtensa LX106 a 80 MHz.
- 64 KiB de RAM per instruccions y 96 KiB de RAM per dades
- IEEE 802.11 b/g/n Wi-Fi
- 16 pins GPIO
- SPI i I2C
- UART als pins dedicats
- Un convertidor Analògic-Digital (ADC) de 10 bits.

Per dur a terme el projecte s'han utilitzat diferents sensors i mòduls externs a la pròpia placa :

- Un display LCD de 20x4 per mostrar les dades a la instal·lació.
- Un sensor DHT11 per fer les mesures.
- Un mòdul *tiny RTC* per mantenir la data i la hora.
- Una protoboard i cables per fer les connexions

## Configuracions

### Configuració de la Raspberry Pi

Per tal de fer funcionar la Raspberry Pi s'han de seguir uns passos. El primer que necessitem és una targeta microSD on guardarem el sistema operatiu. Descarregarem l'eina NOOBS, un programa que ens simplificarà molt la feina de cara a la instal·lació del sistema. Aquest programa inclou un seguit de sistemes operatius per seleccionar el que ens interessi més. En el nostre cas instal·larem Raspbian, una distribució de GNU/Linux basada en Debian, per tant és lliure. Aquest sistema està adaptat pel hardware de la Raspberry. Un cop descarregat i copiat a la microSD introduïrem la targeta a la Raspberry i començarem amb la instal·lació .

Seleccionarem Raspbian, configurarem una connexió a xarxa i polirem els últims detalls com per exemple l'idioma.

## Configuració del Servidor

Un cop tenim la Raspberry Pi en funcionament hem d'instal·lar un seguit de programes i hem de configurar un seguit de paràmetres per tal de poder utilitzar-la com a servidor.

El tipus de servidor que instal·larem s'anomena LAMP, acrònim dels programes que utilitzarem :

- Linux com a sistema operatiu ( en el nostre cas es tracta de la distribució Raspbian.)
- Apache com a servidor web
- Mysql de cara a la gestió de bases de dades
- PHP com a llenguatge de programació .

El primer que hem de fer és actualitzar els repositoris i els programes de la nostra Raspberry pi amb les següents comandes al terminal :

```
sudo apt-get update  
sudo apt-get upgrade
```

Després instal·larem Apache2 :

```
sudo apt-get install apache2
```

Apache és un servidor web de codi obert per plataformes Unix. Windows, Mac i d'altres que implementa el protocol HTTP/1.1. Es tracta del servidor web més utilitzat.

De la mateixa forma instal·larem php7.0 :

```
sudo apt-get install php7.0 libapache2-mod-php7.0
```

Mysql :

```
sudo apt-get install mysql-server mysql-client php7-mysql
```

I PHPMyAdmin :

```
sudo apt-get install phpmyadmin
```

A més, durant la instal·lació de PHPMyAdmin ens demanarà quin tipus de servidor utilitzem (Apache) i que creem les credencials per accedir-hi.

PHPMyAdmin és una eina escrita en PHP que ens permet controlar les bases de dades de MySql de forma senzilla.

Per poder accedir des de fora de casa és necessari que el servidor tingui una IP fixada, és a dir, evitar el DNS. L'alternativa aplicada ha estat utilitzar un servidor de DDNS anomenat *NoIp*. Això ens permet donar un nom de domini a la IP fixa del nostre router, de manera que podrem accedir-hi en remot. En el nostre cas s'anomenarà ***projectefinal.ddns.net***.

## Configuració del Router

Per poder accedir a la web és necessari que el nostre router permeti l'accés de connexions TCP pel port 80, que és el port on està Apache escoltant a l'espera d'enviar les dades del nostre site.

Per fer això ens hem de connectar al router (normalment es troba a la ip 192.168.1.1) i haurem de crear una regla de ports que permeti aquesta connexió :

Editar

Nombre	Protocolo	Puerto/Rango Externo	Puerto/Rango Interno	Direccion IP	Activar
tfg	TCP	6784:6784	6784:6784	192.168.1.60	OFF
web	TCP	80:80	80:80	192.168.1.60	ON
web	UDP	80:80	80:80	192.168.1.60	OFF

1/1

Figura 2 : Ports del Router

Com podem veure hem obert el port 80 per connexions TCP. A més hem apuntat aquestes connexions cap a la direcció IP 192.168.1.60, que és la de la Raspberry Pi. D'aquesta manera, quan intentem accedir des d'una xarxa externa, el router ens derivarà al port 80 de la Raspberry i Apache ens enviarà les dades de la web.

## Site Web

El disseny de la web es basarà en dues pàgines, una on es tindrà accés a la taula amb les dades de les mesures i una altra on hi haurà un formulari de contacte.

A més, abans de poder accedir-hi ens apareixerà una pàgina que ens demanarà una contrasenya per evitar connexions indesitjades. Aprofundirem en això més endavant.

El codi sencer es pot trobar a l'annex, juntament amb els arxius css (arxius que s'encarreguen de donar el format als elements de la pàgina). Seguidament destacaré les parts més importants.

El primer que trobem, com hem comentat unes línies amunt, és un formulari que ens demana una contrasenya :



*Figura 3: Pàgina de Login*

Per fer això s'ha utilitzat el següent codi PHP :

```

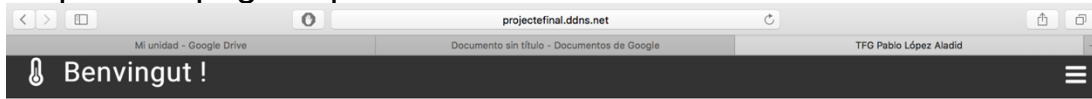
1. <?php
2. session_start(); // Obrim la sessió
3. $password = "b77ee3088c54c6a3de77b9006024f80a";
4. if($_POST['password']){
5.     if(md5($_POST['password']) == $password){
6.         $_SESSION['password'] = "alm";
7.     }else{
8.         echo "<span style='color:red;font-weight:bold;'>La
contrassenya és incorrecta</span>";
9.     }
10.    if(!$_SESSION['password']){
11.        ?>
12.        <link rel="stylesheet" href="css/estilos.css">
13.        <h2>Introdueix La Contrasenya</h2>
14.        <form name="form1" method="post"
action="<?php echo $_SERVER['PHP_SELF']; ?>">
15.        <input type="password" name="password">
16.        <input type="submit" name="Submit"
value="Login!"></form>
17.        <?php
18.        }else{
19.            if($_GET['desconectar']){
20.                session_destroy();
21.                exit("<span style='color:green;'>Desconnexió
realitzada correctament</span>");
22.            }
23.            ?>
24.
25.            //CODI DE LA PÀGINA
26.
27.            <a
href="<?php echo $_SERVER['PHP_SELF']; ?>?desconectar=si">Des
connexió</a>
28.            <?php
29.            }
30.            ?>

```

El que fem és crear una sessió per evitar que se'ns demani la contrassenya. Per comprovar que la contrassenya és correcta s'utilitza un condicional *if*, que comprova si la contrassenya escrita és igual a la establerta.

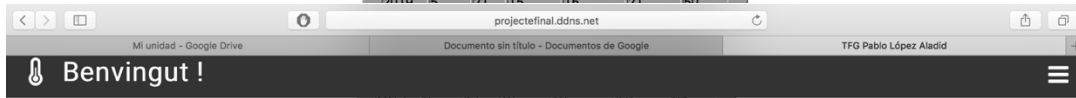
Un cop ens acreditem amb la contrassenya, s'ens mostrarà la pàgina (*en aquest tros de codi no s'ha afegit la pàgina com a tal per evitar que fos massa llarg. Pot trobar-se al final del treball*) ens quedarem dins fins que fem click sobre el botó de desconnexió situat a la part baixa de la web.

La primera pàgina queda així :



Treball Final de Grau Pablo López Aladid

ANY	MES	DIA	HORA	MINUT	TEMP	HUM
2019	5	21	13	34	22.03	46
2019	5	21	13	39	22.02	46
2019	5	21	13	44	21.06	47
2019	5	21	13	49	21.05	46
2019	5	21	13	54	21.05	46
2019	5	21	13	59	21.06	47
2019	5	21	14	4	21.05	48
2019	5	21	14	9	21.06	47
2019	5	21	14	15	21.06	47
2019	5	21	14	20	21.05	47
2019	5	21	14	25	21.05	47
2019	5	21	14	30	21.04	47
2019	5	21	14	35	21.02	48
2019	5	21	14	40	21.01	49
2019	5	21	14	45	21	49
2019	5	21	14	50	21	50
2019	5	21	14	56	21	51
2019	5	21	15	1	20.09	50
2019	5	21	15	6	21	50
2019	5	21	15	11	21	50
2019	5	21	15	16	21	50



2019	5	21	18	42	20	53
2019	5	21	18	47	20.01	53
2019	5	21	18	52	20.01	54
2019	5	21	18	57	20.01	54
2019	5	21	19	3	20.01	54
2019	5	21	19	8	20.01	54
2019	5	21	19	13	20.01	55
2019	5	21	19	18	20.01	55
2019	5	21	19	23	20.01	55
2019	5	21	19	28	20.01	55
2019	5	21	19	34	20.01	55
2019	5	21	19	39	20.01	56
2019	5	21	19	44	20.01	56
2019	5	21	19	49	20.01	56
2019	5	21	19	54	20.01	56
2019	5	21	19	59	20.01	57
2019	5	21	20	4	20.02	58
2019	5	21	20	9	20.03	58
2019	5	21	20	15	20.03	58
2019	5	21	20	20	20.04	57
2019	5	21	20	25	20.06	58
2019	5	21	20	30	20.07	58
2019	5	21	20	35	20.07	58

[Desconexió](#)

Figures 4 i 5 : Primera pàgina del site

Com podem veure, la pàgina s'estructura en un banner superior, la taula de resultats i un cercador que ens permet buscar les entrades per mes :





DIA , HORA, TEMPERATURA I HUMITAT

21 17 :9 20.02 52

21 18 :6 20.01 53

21 18 :11 20.01 53

21 18 :16 20.01 53

21 18 :22 20.01 53

21 18 :27 20.01 53

21 18 :32 20 53

21 18 :37 20 54

21 18 :42 20 53

21 18 :1 20.01 53

21 17 :56 20.01 53

21 17 :15 20 02 51

Figura 6: Resultats del cercador

El codi escrit per aquesta pàgina pot trobar-se al final d'aquest document. Aquí destacarem les parts més rellevants.

```

1.  <!DOCTYPE html>
2.  <html lang ="en">
3.
4.  <head> <!-- Aquí diem on es troben els arxius css i
    especifiquem les característiques de la web -->
5.  <meta charset="UTF-8">
6.  <meta name="viewport" content="width=device-width, user-
    scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1">
7.  <link rel="stylesheet" href="http://projectefinal.ddns.net/cs
    s/fontello.css">
8.
9.
10. <link rel="stylesheet" href="http://projectefinal.ddns.n
    et/css/estilos.css">
11. <title> TFG Pablo López Aladid </title>
12.
13. </head>
14.
15. <body>
16.
17. <header> <!-- Creem un espai a la part superior-->
18. <div class="contenedor"> <!-- Dins d'aquest espai
    carreguem una icona d'un termòmetre i escrivim "Benvingut!"-->
19. <h1 class="icon-temperature"> Benvingut ! </h1>
20. <input type="checkbox" id="menu-bar">
21. <label class="icon-menu" for="menu-bar"></label>

```

```

22.         <nav class="menu"> <!-- A més. afegim un menú que farem
desplegable amb css-->
23.         <a href="index.php">Inici</a>
24.         <a href="informacion.php">Informació i Contacte</a>
25.     </nav>
26. </div>
27. </header>
28. <main>
29.     <section id="banner">
30.         30.  <!-- Afegim el logo de la
UPC amb el títol -->
31.
32.         <h2> Treball Final de Grau Pablo López Aladid</h2>
33.
34.     </section>
35.
36.     <!-- Obrim codi PHP per llegir la base de dades -->

```

Aquesta part és comuna entre les dues pàgines i el que fem és indicar on es troben els documents css i definim les característiques principals de la pàgina. A més, creem un menú que farem desplegable més tard amb css.

Després, amb PHP, mostrem les dades i creem el cercador per mesos:

```

1. <?php
2. $dbhost = "localhost";
3. $dbuser = "admin";
4. $dbpass = "admin";
5. $dbname = "definitivo";
6.
7. if($con = mysqli connect($dbhost, $dbuser, $dbpass, $dbn
ame)){ // si la connexió és satisfactòria carreguem les
dades i les mostrem per pantalla
8.
9. $sentencia = "SELECT * FROM lecturas";
10.     $resultado = mysqli query($con, $sentencia);
11.     if($resultado)
12.
13.         echo '<table border="1" width = "100%" bgcolor =
"#b2b2b4">'; // per mostrar les dades creem una taula i
la omplim
14.         echo '<td>' . "ANY" . '</td><td>' . "MES" . '</td><td
>' . "DIA" . '</td><td>' . "HORA" . '</td><td>' . "MINUT" . '<
/td><td>' . "TEMP" . '</td><td>' . "HUM" . '</td>';
15.         while($fila = mysqli fetch array($resultado))
16.         {
17.             echo '<tr>';
18.             echo '<td>' . $fila['any'] . '</td><td>' . $fila['m
es'] . '</td><td>' . $fila['dia'] . '</td><td>' . $fila['h

```

```

ora'] . '</td><td>' . $fila['minut'] . '</td><td>' . $fila
['temp'] . '</td><td>' . $fila['hum'] . '</td>';
19.     echo '</tr>';
20.     }
21.     echo '</table>';
22.
23.
24.     }
25.
26.
27.     $texto = '';
28.
29.     $registros = '';
30.
31.     if($_POST){
32.
33.         $busqueda = trim($_POST['buscar']);
34.
35.         $entero = 0;
36.
37.         if (empty($busqueda)){
38.             $texto = '';
39.             echo $texto.<br>';
40.         }else{
41.             //Si hi ha informació per buscar obrim la connexió
42.
43.             mysqli set charset('utf8');
44.
45.             //Consulta per la base de dades
46.             $sql = "SELECT * FROM lecturas WHERE mes LIKE
'%".$busqueda."%' ORDER BY dia";
47.
48.             $resultado2 = mysqli query($con, $sql); //Execució
de la consulta
49.             //Si trobem resultats...
50.             if (mysqli num rows($resultado2) > 0){
51.                 $texto = "DIA , HORA, TEMPERATURA I HUMITAT";
52.                 echo $registros;
53.                 echo $texto.<br>';
54.                 //Guardem el nombre de resultats
55.                 $registros = '<p>HEM TROBAT
' . mysqli num rows($resultado2) . ' registres </p>';
56.                 //Guardem les cadenes de resultats
57.                 while($fila = mysqli fetch assoc($resultado2)){
58.
59.                     $dia = $fila['dia'];
60.                     $hora = $fila['hora'];
61.                     $minut = $fila['minut'];

```

```
62.     $temp = $fila['temp'];
63.     $hum = $fila['hum'];
64.     echo $dia ." " . $hora."
: " . $minut ." " . $temp ." " . $hum."</br>";
65.     }
66.
67.     }else{
68.         $texto = "SENSE RESULTATS"; // si no trobem cap
        resultat ho direm
69.         echo $texto."</br>";
70.     }
71.
72.     }
73.     }
74.
75.
76.     mysqli_close($con);
77.
78.     ?>
```

Com podem veure a les primeres línies, obrim una connexió amb la funció *mysqli\_connect*. Després amb una sentència SQL seleccionem totes les dades de la base i les mostrem en forma de taula (de la línia 9 a la 24). Un cop acabem de mostrar les dades construïm el cercador per mesos. El que fem és crear una variable pel mes que volem buscar i amb una sentència SQL similar a la d'abans fem la consulta a la base de dades. En aquest cas en lloc de mostrar totes les dades mostrarem aquelles on el mes sigui igual al que s'ha introduït (línia 46). Finalment imprimim per pantalla les dades.

La segona pàgina no té la taula de mesures. En el seu lloc podem trobar un formulari de contacte ( està deshabilitat per evitar rebre correus ara mateix) :

Desconnexió

Figura 7: Segona pàgina de Site

El que hem fet ha estat canviar el codi de la taula pel del formulari, deixant la resta de la pàgina igual, incloent la part d'inici de sessió. Podem veure com apareix el botó de "Desconnexió" a la part inferior. Per fer el formulari s'ha utilitzat el següent codi :

```
1. <form action="">
2.   <h2>CONTACTE</h2>
3.   <input type="text" name="Nom" placeholder="Nom">
4.   <input type="text" name="Correu" placeholder="Correu">
5.   <textarea name="missatge" placeholder="Escriu un missatge..."></textarea>
6.   <input type="button" value="Enviar" id="boton">
7. </form>
```

Creem un formulari amb 3 *inputs*, que són el nom i el correu de la persona interessada en contactar i el cos del missatge.

## Programació de la placa Arduino

Pel que fa el codi, el que s'ha fet ha estat provar els mòduls per separat i, un cop s'han fet funcionar tots individualment, s'han anat mesclant fins arribar al disseny final.

Podeu trobar el codi sencer a l'annex, aquí es comentaran les parts més rellevants.

Per començar, les llibreries utilitzades han estat les següents:

```
**/  
#include <DHT11.h>  
#include <LiquidCrystal_I2C.h>  
#include "RTCLib.h"  
#include <Wire.h>  
#include<ESP8266WiFi.h>
```

La primera s'encarrega de gestionar la comunicació amb el sensor de temperatura i humitat. Aquest sensor, anomenat DHT11 és digital, de forma que estem més protegits davant el soroll que amb un analògic.



Figura 8: Sensor DHT11

La segona llibreria és per controlar el display LCD i la tercera serveix per emprar el mòdul *TinyRTC*. Un mòdul RTC és un rellotge digital que podem connectar a un microprocessador. Per les seves sigles en anglès podem dir que és un *Real Time Clock*.

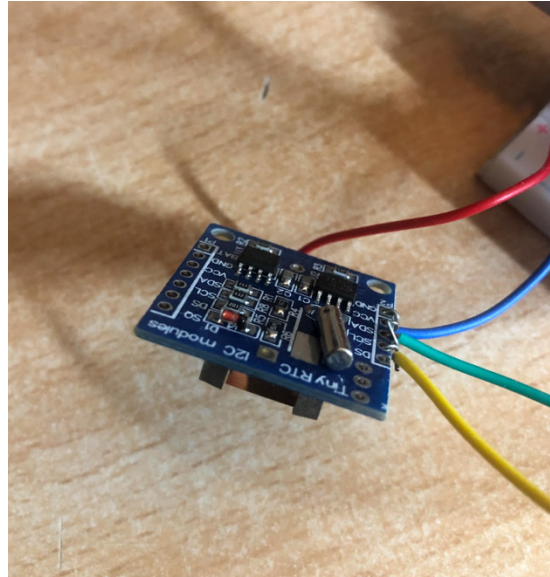


Figura 9: Mòdul TinyRTC

La quarta ens ajuda amb la comunicació amb dispositius I2C (pantalla LCD) i finalment la última és l'encarregada d'obrir la comunicació via WIFI del mòdul integrat a la placa.

El següent pas és declarar una sèrie de variables que utilitzarem més endavant :

```
const char* ssid = "-----"; // nom de la xarxa
const char* pass = "-----"; // clau wifi
const char* host = "projectefinal.ddns.net" ;

int id = 1; // és la variable que identifica les mesures a la base de dades
int pin = 14; // Aquest és el pin on connectarem el sensor
DHT11 dht11(pin); // obrim una instància de DHT11

LiquidCrystal_I2C lcd(0x27, 20, 4); // definim la pantalla lcd
RTC_DS1307 RTC; // definim el tinyrtc
```

Les tres primeres variables són les que passarem més endavant a la funció de wifi (les dues primeres han estat modificades ja que són les credencials d'accés a la xarxa de casa meva). La *id* ens serveix per identificar les mesures a la base de dades i *pin* ens serveix per dir que el sensor es connectarà al pin 14 de la NodeMCU.

```
Serial.begin(9600);

//delay(5000);
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid,pass);
while (WiFi.status() != WL_CONNECTED) { // imprimirem un punt mentres esperem a la connexió
  delay(500);
  Serial.print(".");
}
delay(3000);
Serial.println(); // quan es connecti mostrarem la IP assignada
Serial.println("WiFi Connected");
Serial.println("IP Address : ");
Serial.println(WiFi.localIP());
```

En aquest fragment de codi podem destacar l'ús de la llibreria wifi. Ens serveix per connectar-nos a la xarxa passant-li els paràmetres de *ssid* i la contrasenya. Un cop s'estableix la connexió mostrem pel monitor sèrie la IP que rep l'Arduino.

```
Wire.begin();
RTC.begin();
if (! RTC.isrunning()) {
  Serial.println("RTC is NOT running!");
  RTC.adjust(DateTime(__DATE__, __TIME__));
}

lcd.begin(20, 4);
lcd.init();
lcd.backlight();
lcd.clear();
}
```



A l'última part del setup inicialitzem el rellotge amb la data i hora actuals i el display lcd.

La primera part del loop no és de gran interès atès que es tracta de declaracions de variables i impressions al display:

```
void loop()
{
    DateTime now = RTC.now();

    int any = now.year(), mes = now.month(), dia = now.day(), hora = now.hour(), minut = now.minute();

    int err;
    float temp, hum;
    if ((err = dht11.read(hum, temp)) == 0) // Si devuelve 0 es que ha leído bien
    {
        Serial.print("Temp: ");
        Serial.print(temp);
        lcd.setCursor(0, 0);
        lcd.print("Temp: ");
        lcd.print(temp);
        Serial.print("Hum: ");
        Serial.print(hum);
        lcd.setCursor(0, 1);
        lcd.print("Hum: ");
        lcd.print(hum);
        Serial.println();

        lcd.setCursor(0, 2);
        lcd.print(now.day(), DEC);
        lcd.print('/');
        lcd.print(now.month(), DEC);
        lcd.print('/');
        lcd.print(now.year(), DEC);
    }
}
```

La part més important és la següent:

```
WiFiClient client;
const int httpPort = 80;
if(!client.connect(host,httpPort)) {
    Serial.println("Connection failed");
    return;
}

String url = "/conexion.php";
String key = "?pass=admin";
String dato1 = "&id=";
String dato2 = "&temp=";
String dato3 = "&hum=";
String dato4 = "&any=";
String dato5 = "&mes=";
String dato6 = "&dia=";
String dato7 = "&hora=";
String dato8 = "&minut=";
```

Veiem com declarem una variable anomenada *port* que ens serveix per dir a la funció *client.connect* a quin port del host (definit prèviament a la primera imatge) ens connectarem. També declarem la variable *url*, que

ens serveix per apuntar a l'arxiu **conexion.php**. Aquest arxiu conté les dades per realitzar la connexió a la base de dades i el veurem amb més profunditat més endavant. Després declarem un seguit de *strings* que ens serviran per muntar la sentència que pujarà les dades al servidor:

```
Serial.print("Requesting URL: ");
Serial.println(url);
client.print(String("GET ") + url + key + dato1 + id + dato2 + temp +
dato3 + hum + dato4 + any +
dato5 + mes + dato6 + dia + dato7 + hora + dato8 + minut + "HTTP/1.1\r\n" +
"Host: " + host + "\r\n" +
"Connection: close\r\n\r\n");
```

Per evitar que la connexió quedi oberta massa temps definim un *timeout* de 5 segons. Si es passa aquest temps sense resposta la connexió es tancarà :

```
unsigned long timeout = millis();
while(client.available() == 0){
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout");
    client.stop();
    return;
  }
}
```

Finalment imprimim pel monitor sèrie la resposta del servidor :

```
while(client.available()) {
  String line = client.readStringUntil('\r');
  Serial.print(line);
}
Serial.println();
Serial.println("closing connection");
id=id+1;
delay(300000);
}
```

## Base de dades

Per crear la base de dades a PHPMyAdmin podem fer-ho de diferents formes. La forma manual és més simple si no es tenen coneixements de SQL, però també és més lenta. La solució proposada, doncs, és crear un script amb sentències SQL i carregar l'arxiu a PHPMyAdmin. Un cop fet això s'executaran les sentències i es crearà la base de dades. En el nostre cas la base s'anomena *definitivo* i la taula on guardarem les dades s'anomena *lecturas*. Aquesta taula consta de diferents columnes :

- ID : és l'identificador de la mesura
- Any : any de la mesura
- Mes : mes de la mesura
- Dia : dia de la mesura
- Hora : hora de la mesura
- Minut : minut de la mesura
- Temp : temperatura mesurada
- Hum : humitat mesurada

Llavors, l'arxiu *CreaDatabase* queda així

```
1. CREATE DATABASE definitivo;
2. USE definitivo;
3. CREATE TABLE lecturas (
4. id INT NOT NULL AUTO_INCREMENT,
5. any INT NOT NULL,
6. mes INT NOT NULL,
7. dia INT NOT NULL,
8. hora INT NOT NULL,
9. minut INT NOT NULL,
10.    temp FLOAT NOT NULL,
11.    hum FLOAT NOT NULL,
12.    PRIMARY KEY(id)
13. );
14.
15.
```

## Circuit de l'estació meteorològica

Com hem comentat, la placa NodeMCU s'utilitza amb diferents components externs per aconseguir la nostra finalitat. A continuació es pot veure un esquema de les connexions realitzades :

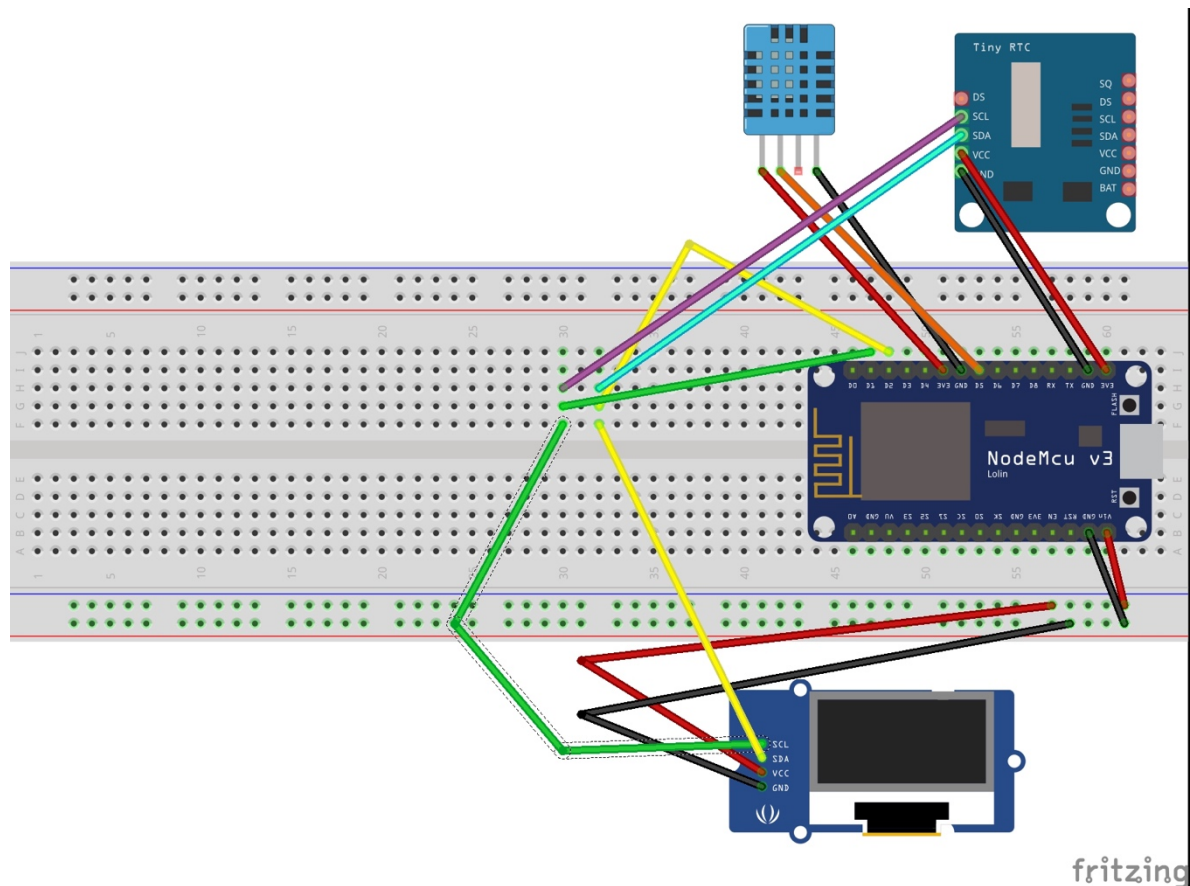


Figura 10: Esquema de connexions

Podem veure que tenim la NodeMCU al centre, el display LCD a la part baixa i el sensor dht11 i el mòdul tinyRTC a la part superior.



El muntatge ha quedat així :

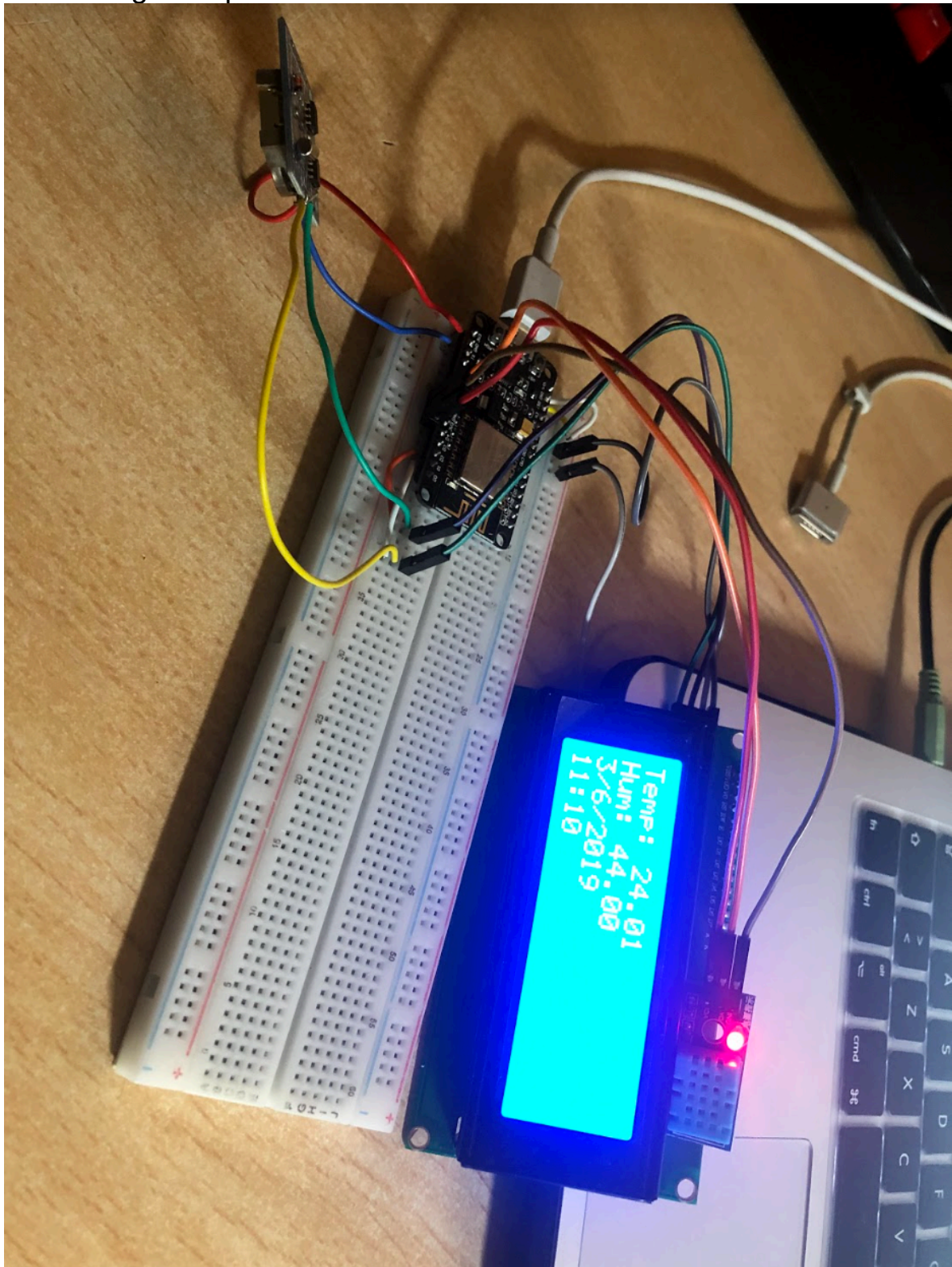


Figura 11: Muntatge

## Seguretat implementada al Site

Per evitar connexions no desitjades a la nostra web s'ha implementat un sistema d'autenticació mitjançant un *password* o contrasenya. Per fer-ho s'ha escrit un codi PHP que ens demana que fem *login* abans de presentar-nos la web. En aquest codi s'ha aprofitat la funció `md5()` de PHP, que ens permet encriptar la contrasenya de forma senzilla.

### Què és la encriptació MD5 ?

MD5 son les sigles de *Message-Digesting Algorhythm 5*, cosa que podríem traduir com Algoritme de Resum de Missatge 5. Es tracta d'un algoritme de reducció criptogràfic dissenyat pel professor Ronald Rivest al *Massachusetts Institute of Technology*. És una codificació de 128 bits, representada normalment per 32 caràcters hexadecimal. Un exemple del que fa és el següent :

`MD5( 'contrasenya' ) = 82af7fd93573df7b229a3867c80fd72f`

`MD5( 'contrasanya' ) = 30f8bf79bd5a4c226f117b3feddb98bb`

Podem veure, doncs, que només canviant una lletra de la paraula (en aquest cas una 'e' per una 'a') la sortida és radicalment diferent.

El procés de codificació és massa extens i queda fora de l'abast d'aquest treball. No obstant, es citaran els passos que segueix l'algoritme per arribar a la sortida :

- Primer s'afegeixen bits al missatge
- Després s'afegeix un enter de 64 bits que representa la longitud del missatge
- S'inicialitza un buffer de 4 paraules (A,B,C,D) de 32 bits cadascuna.
- Es processa el missatge en blocs de 16 paraules
- La sortida és la combinació de A,B,C i D.

## Connexió a la base de dades

### Config.php

Aquest arxiu ens serveix per crear la variable de connexió a la base de dades que passarem a l'arxiu *conexion.php*.

```
1.<?php
2.$dbhost = "localhost";
3.$dbuser = "admin";
4.$dbpass = "admin";
5.$dbname = "definitivo";
6.
7.$con = mysqli_connect($dbhost, $dbuser, $dbpass,
    $dbname);
8.?>
```

Simplement diem on es troba el host (en aquest cas és el propi ordinador, per això posem localhost), passem les credencials d'usuari i contrasenya i el nom de la base de dades. Després creem la variable *con*, que és la que obrirà la connexió.

## Conexion.php

L'arxiu *conexion.php* ens permet connectar la placa NodeMCU a la base de dades per poder pujar-hi les lectures del sensor dht11. El codi d'aquest arxiu és el següent :

```

1. <?php
2. require("config.php"); //demanem l'arxiu config.php
3. $id = mysqli_real_escape_string($con, $_GET['id']);
4. $temp = mysqli_real_escape_string($con, $_GET['temp']);
5. $hum= mysqli_real_escape_string($con, $_GET['hum']);
6. $any= mysqli_real_escape_string($con, $_GET['any']);
7. $mes= mysqli_real_escape_string($con, $_GET['mes']);
8. $dia= mysqli_real_escape_string($con, $_GET['dia']);
9. $hora= mysqli_real_escape_string($con, $_GET['hora']);
10.     $minut= mysqli_real_escape_string($con, $_GET['minu
    t']);
11.     $query = "INSERT INTO lecturas(id, any, mes, dia,
    hora, minut, temp, hum) VALUES('$id'
    , '$any', '$mes', '$dia', '$hora', '$minut', '$temp',
    '$hum')";
12.     mysqli_query($con, $query);
13.     mysqli_close($con);
14.     echo "Pagina para subir los datos<br />";
15.     echo "<br />id = $id <br />";
16.     echo "<br />temp = $temp <br />";
17.     echo "<br />hum = $hum <br />";
18.     echo "<br />any = $any <br />";
19.     echo "<br />mes = $mes <br />";
20.     echo "<br />dia = $dia <br />";
21.     echo "<br />hora = $hora <br />";
22.     echo "<br />minut = $minut <br />";
23.     ?>

```



Aquest codi crea una variable per cada columna de la base de dades. (Els valors d'aquestes variables els passa la placa NodeMCU). Després creem una sentència SQL per insertar les dades rebudes a la taula *lecturas* de la base de dades anomenada *definitivo*.

## Pressupost

Per calcular el pressupost s'ha fet la suma dels costos dels materials emprats, de les hores de treball (s'ha suposat un salari d'enginyer en pràctiques de 8€ la hora i un total de 300 hores). A més, s'ha afegit el 21% en concepte de IVA, un 20% de benefici industrial i un 20% en concepte de costos d'estructura:

Concepte	Cost	Quantitat	Sub-Total
Raspberry Pi 3b+ i accessoris	72,99 €	1,00 €	72,99 €
NodeMCU esp8266	7,99 €	1,00 €	7,99 €
Sensor dht11	2,79 €	1,00 €	2,79 €
TinyRTC	1,60 €	1,00 €	1,60 €
Display LCD	6,99 €	1,00 €	6,99 €
Cablejat	0,87 €	1,00 €	0,87 €
Protoboard	8,99 €	1,00 €	8,99 €
Hores de Treball	8,00 €	300,00 €	2.400,00 €
		Sub-Total	2.502,22 €
		IVA(%)	21
		Benefici Ind.(%)	20
		Estructura(%)	20
		Total	3.503,11 €
		Total + IVA	4.238,76 €

Figura 12: Càlculs del pressupost

Com podem veure, el total una vegada aplicats els impostos corresponents és de 4238,76€.

## Conclusions i accions futures

Un cop acabat el treball podem dir que els resultats han estat satisfactoris, ja que s'han assolit els objectius establerts. S'ha aconseguit muntar un servidor funcional amb una web allotjada, i s'ha aconseguit que aquesta web mostri per pantalla les mesures obtingudes amb el circuit dissenyat amb la placa Arduino.

Com accions futures es pot intentar redissenyar el circuit per afegir-hi una font d'alimentació i així evitar haver de tenir el circuit connectat a un port USB d'ordinador.

També es podrien afegir més sensors per poder obtenir més dades. Finalment, podria afegir-se un apartat estadístic a la pàgina web.

## Codis sencers

### Primera pàgina del Site

```

79.     <?php
80.         session_start(); // obrim la sessió
81.         $password = "b77ee3088c54c6a3de77b9006024f80
a"; //declarem la contrasenya en md5
82.         if($_POST['password']){ //validem la
contrasenya
83.             if(md5($_POST['password']) == $password){
84.                 $_SESSION['password'] = "alm"; // si és
correcta permetem la connexió
85.             }else{
86.                 echo "<span style='color:red;font-
weight:bold;'>La contrassenya és
incorrecta</span>"; // si no és correcta avisem
amb un missatge vermell
87.             }}
88.             if(!$_SESSION['password']){
89.                 ?>
90.                 <link rel="stylesheet"
href="css/estilos.css">
91.                 <h2>Introdueix La Contrasenya</h2>
92.                 <form name="form1" method="post"
action="<?php echo $_SERVER['PHP_SELF']; ?>">
93.                     <input type="password" name="password">
94.                     <input type="submit" name="Submit"
value="Login!"></form>
95.                 <?php
96.                 }else{
97.                     if($_GET['desconectar']){ // si fem click a
desconectar tanquem la sessió
98.                         session_destroy();
99.                         exit("<span style='color:green;'>Desconnexió
realitzada correctament</span>");
100.                    }
101.                    ?>

```

```
102. <!DOCTYPE html>
103. <html lang ="en">
104.
105. <head> <!-- Aquí diem on es troben els
      arxius css i especifiquem les característiques
      de la web -->
106. <meta charset="UTF-8">
107. <meta name="viewport" content="width=device-
      width, user-scalable=no, initial-scale=1,
      maximum-scale=1, minimum-scale=1">
108. <link rel="stylesheet"
      href="http://projectefinal.ddns.net/css/fontello
      .css">
109.
110.
111. <link rel="stylesheet"
      href="http://projectefinal.ddns.net/css/estilos.
      css">
112. <title> TFG Pablo López Aladid </title>
113.
114. </head>
115.
116. <body>
117.
118. <header> <!-- Creem un espai a la part
      superior-->
119. <div class="contenedor"> <!-- Dins d'aquest
      espai carreguem una icona d'un termòmetre i
      escrivim "Benvingut!"-->
120. <h1 class="icon-temperature"> Benvingut !
      </h1>
121. <input type="checkbox" id="menu-bar">
122. <label class="icon-menu" for="menu-
      bar"></label>
123. <nav class="menu"> <!-- A més. afegim un
      menú que farem desplegable amb css-->
124. <a href="index.php">Inici</a>
125. <a href="informacion.php">Informació i
      Contacte</a>
126. </nav>
```

```

127. </div>
128. </header>
129. <main>
130. <section id="banner">
131.  <!-- Afegim el
    logo de la UPC amb el títol -->
132.
133. <h2> Treball Final de Grau Pablo López
    Aladid</h2>
134.
135. </section>
136.
137. <!-- Obrim codi PHP per llegir la base de
    dades -->
138.
139. <?php
140. $dbhost = "localhost";
141. $dbuser = "admin";
142. $dbpass = "admin";
143. $dbname = "definitivo";
144.
145. if($con = mysqli connect($dbhost, $dbuser, $d
    bpass, $dbname)){ // si la connexió és
    satisfactòria carreguem les dades i les mostrem
    per pantalla
146.
147. $sentencia = "SELECT * FROM lecturas";
148. $resultado = mysqli query($con, $sentencia);
149. if($resultado)
150.
151. echo '<table border="1" width = "100%"
    bgcolor = "#b2b2b4">'; // per mostrar les dades
    creem una taula i la omplim
152. echo '<td>' . "ANY" . '</td><td>' . "MES". '<
    /td><td>' . "DIA". '</td><td>' . "HORA". '</td><td
    >' . "MINUT". '</td><td>' . "TEMP". '</td><td>' .
    "HUM". '</td>';
153. while($fila = mysqli fetch array($resultado))
154. {
155. echo '<tr>';

```

```

156.     echo '<td>' . $fila['any'] . '</td><td>' . $f
        ila['mes'] . '</td><td>' . $fila['dia'] . '</td><td>'
        . $fila['hora'] . '</td><td>' . $fila['minut'] . '
        </td><td>' . $fila['temp'] . '</td><td>' . $fila['
        hum'] . '</td>';
157.     echo '</tr>';
158.     }
159.     echo '</table>';
160.
161.
162.     }
163.
164.
165.     $texto = '';
166.
167.     $registros = '';
168.
169.     if($_POST){
170.
171.         $busqueda = trim($_POST['buscar']);
172.
173.         $entero = 0;
174.
175.         if (empty($busqueda)){
176.             $texto = '';
177.             echo $texto.'<br>';
178.         }else{
179.             //Si hi ha informació per buscar obrim la
                connexió
180.
181.             mysqli set charset('utf8');
182.
183.             //Consulta per la base de dades
184.             $sql = "SELECT * FROM lecturas WHERE mes
                LIKE '%" . $busqueda . "%' ORDER BY dia";
185.
186.             $resultado2 = mysqli_query($con, $sql); //Exe
                cució de la consulta
187.             //Si trobem resultats...

```

```

188.  if (mysqli_num_rows($resultado2) > 0){
189.  $texto = "DIA , HORA, TEMPERATURA I
      HUMITAT";
190.  echo $registros;
191.  echo $texto.'mysqli_num_rows($resultado2) . ' registres
      </p>';
194.  //Guardem les cadenes de resultats
195.  while($fila = mysqli_fetch_assoc($resultado2)
      ){
196.
197.  $dia = $fila['dia'];
198.  $hora = $fila['hora'];
199.  $minut = $fila['minut'];
200.  $temp = $fila['temp'];
201.  $hum = $fila['hum'] ;
202.  echo $dia ." " . $hora."
      :". $minut." " . $temp." " . $hum.'mysqli_close($con);
215.
216.  ?>
217.
218.  <!-- Creem el buscador a la pàgina-->

```



```

219. <form id="buscador" name="buscador"
    method="post"
    action="<?php echo $_SERVER['PHP_SELF'] ?>">
220. <input id="buscar" name="buscar"
    type="search" placeholder="Buscar un mes"
    autofocus >
221. <input type="submit" name="buscador"
    class="boton" value="buscar">
222. </form>
223.
224. </main>
225.
226. </body>
227.
228. </html>
229. <a
    href="<?php echo $_SERVER['PHP_SELF']; ?>?descone
    ctar=si">Desconnexió</a>
230. <?php
231. }
232. ?>

```

## Segona pàgina del Site

```

1.<?php
2.session_start();
3.$password = "b77ee3088c54c6a3de77b9006024f80a";
4.if($_POST['password']){
5.if(md5($_POST['password']) == $password){
6.$_SESSION['password'] = "alm";
7.}else{
8.echo "<span style='color:red;font-
   weight:bold;'>La contrassenya és
   incorrecta</span>";
9.}}
10.    if(!$_SESSION['password']){
11.        ?>
12.        <link rel="stylesheet"
   href="css/estilos.css">
13.        <h2>Introdueix La Contrasenya</h2>
14.        <form name="form1" method="post"
   action="<?php echo $_SERVER['PHP_SELF']; ?>">
15.        <input type="password" name="password">
16.        <input type="submit" name="Submit"
   value="Login!"></form>
17.        <?php
18.        }else{
19.        if($_GET['desconectar']){
20.        session_destroy();
21.        exit("<span style='color:green;'>Desconnexió
   realitzada correctament</span>");
22.        }
23.        ?>
24.        <!DOCTYPE html>
25.        <html lang = "en">
26.
27.        <head>
28.        <meta charset="UTF-8">

```

```
29.     <meta name="viewport" content="width=device-
      width, user-scalable=no, initial-scale=1,
      maximum-scale=1, minimum-scale=1">
30.     <link rel="stylesheet"
      href="css/fontello.css">
31.     <link rel="stylesheet"
      href="css/estilos.css">
32.     <title> TFG Pablo López Aladid </title>
33.
34.     </head>
35.
36.     <body>
37.
38.     <header>
39.     <div class="contenedor">
40.     <h1 class="icon-temperature"> Benvingut !
      </h1>
41.     <input type="checkbox" id="menu-bar">
42.     <label class="icon-menu" for="menu-
      bar"></label>
43.     <nav class="menu">
44.     <a href="index.php">Inici</a>
45.     <a href="informacion.php">Informació i
      Contacte</a>
46.     </nav>
47.     </div>
48.     </header>
49.     <main>
50.     <section id="banner">
51.     
52.     <h2> Treball Final de Grau Pablo López
      Aladid</h2>
53.     <br/>
54.     <!--La diferència entre aquesta pàgina i la
      primera és que en aquesta no mostrem les dades
      de la base, sinó que trobem un formulari de
      contacte. En aquest cas no està actiu per evitar
      que m'arribin correus. -->
55.     </section>
56.     <br>
```

```

57.     <h3> Aquesta és la web del meu treball final
        de grau en enginyeria de sistemes
        audiovisuals</h3>
58.     <br/>
59.     <form action="">
60.     <h2>CONTACTE</h2>
61.     <input type="text" name="Nom"
        placeholder="Nom">
62.     <input type="text" name="Correu"
        placeholder="Correu">
63.     <textarea name="missatge"
        placeholder="Escriu un missatge..."></textarea>
64.     <input type="button" value="Enviar"
        id="boton">
65.     </form>
66.
67.     </main>
68.
69.
70.     </body>
71.
72.
73.
74.     </html>
75.     <a
        href="<?php echo $_SERVER['PHP_SELF']; ?>?descone
        ctar=si">Desconnexió</a>
76.     <?php
77.     }
78.     ?>

```

## Codi Arduino

```

1. /** Hem d'incloure les llibreries necessàries :
2.- DHT11.h per comunicar-nos amb el sensor de
   temperatura i humitat.
3.- LiquidCrystal_I2C.h per controlar el panell
   LCD. S'escull aquesta llibreria perquè ens
   facilita molt la comunicació.
4.- RTCLib.h per comunicar-nos con el rellotge
   tinyRTC
5.- Wire.h ens serveix per comunicar-nos amb
   mòduls I2C
6.**/
7.#include <DHT11.h>
8.#include <LiquidCrystal_I2C.h>
9.#include "RTCLib.h"
10.    #include <Wire.h>
11.    #include<ESP8266WiFi.h>
12.
13.    const char* ssid = "-----"; // nom de la
   xarxa
14.    const char* pass = "-----"; // clau wifi
15.    const char* host = "projectefinal.ddns.net"
   ;
16.
17.    int id = 1; // és la variable que identifica
   les mesures a la base de dades
18.    int pin = 14; // Aquest és el pin on
   connectarem el sensor
19.    DHT11 dht11(pin); // obrim una instància de
   DHT11
20.
21.    LiquidCrystal_I2C lcd(0x27, 20, 4); //
   definim la pantalla lcd
22.    RTC_DS1307 RTC; // definim el tinyrtc
23.
24.

```

```

25. void setup()
26. {
27.   Serial.begin(9600);
28.
29.   //delay(5000);
30.   Serial.println();
31.   Serial.println();
32.   Serial.print("Connectant a:");
33.   Serial.println(ssid);
34.
35.   WiFi.begin(ssid,pass);
36.   while (WiFi.status() != WL_CONNECTED) {
37.     delay(500);
38.     Serial.print(".");
39.   }
40.   delay(3000);
41.   Serial.println();
42.   Serial.println("WiFi Connectat");
43.   Serial.println("IP Address : ");
44.   Serial.println(WiFi.localIP());
45.
46.
47.   Wire.begin();
48.   RTC.begin();
49.   if (! RTC.isrunning()) {
50.     Serial.println("RTC no s'executa!");
51.     RTC.adjust(DateTime(__DATE__, __TIME__));
52.   }
53.   lcd.begin(20, 4);
54.   lcd.init();
55.   lcd.backlight();
56.   lcd.clear();
57. }
58. void loop()
59. {
60.   DateTime now = RTC.now();
61.

```

```

62.     int any = now.year(),mes = now.month(), dia =
        now.day(), hora = now.hour(), minut = now.minute
        ();
63.
64.
65.     int err;
66.     float temp, hum;
67.     if ((err = dht11.read(hum, temp)) == 0)
68.     {
69.         Serial.print("Temp: ");
70.         Serial.print(temp);
71.         lcd.setCursor(0, 0);
72.         lcd.print("Temp: ");
73.         lcd.print(temp);
74.         Serial.print("Hum: ");
75.         Serial.print(hum);
76.         lcd.setCursor(0, 1);
77.         lcd.print("Hum: ");
78.         lcd.print(hum);
79.         Serial.println();
80.
81.         lcd.setCursor(0, 2);
82.         lcd.print(now.day(), DEC);
83.         lcd.print('/');
84.         lcd.print(now.month(), DEC);
85.         lcd.print('/');
86.         lcd.print(now.year(), DEC);
87.
88.         lcd.setCursor(0, 3);
89.         lcd.print(now.hour(), DEC);
90.         lcd.print(':');
91.         lcd.print(now.minute(), DEC);
92.
93.         delay(2000);
94.     }
95.
96.     Serial.print("Connectant a:");
97.     Serial.println(host);

```

```

98.
99.
100.
101. WiFiClient client;
102. const int httpPort = 80;
103. if(!client.connect(host,httpPort)) {
104.   Serial.println("Connexió fallida ");
105.   return;
106. }
107. String url = "/conexion.php";
108. String key = "?pass=admin";
109. String dato1 = "&id=";
110. String dato2 = "&temp=";
111. String dato3 = "&hum=";
112. String dato4 = "&any=";
113. String dato5 = "&mes=";
114. String dato6 = "&dia=";
115. String dato7 = "&hora=";
116. String dato8 = "&minut=";
117.
118. Serial.print("URL: ");
119. Serial.println(url);
120. client.print(String("GET
    ") + url + key + dato1 + id + dato2 + temp + dat
    o3 + hum + dato4 + any + dato5 + mes + dato6 + d
    ia + dato7 + hora + dato8 + minut + "HTTP/1.1\r\
    n" +
121.   "Host: " + host + "\r\n" +
122.   "Connection: close\r\n\r\n");
123.
124. unsigned long timeout = millis();
125. while(client.available() == 0){
126.   if (millis() - timeout > 5000) {
127.     Serial.println(">>> Client Timeout");
128.     client.stop();
129.     return;
130.   }
131. }

```



```
132.
133. while(client.available()) {
134.   String line = client.readStringUntil('\r');
135.   Serial.print(line);
136. }
137. Serial.println();
138. Serial.println("tancant connexió");
139. id=id+1;
140. delay(300000);
141. }
```

## Codi CSS

Estilos.css

```
1. @import url('https://fonts.googleapis.com/css?family=Roboto')
;
2. @import url(menu.css);
3.
4.
5. *{
6.     margin: 0;
7.     padding: 0;
8.     box-sizing: border-box;
9. }
10.
11. body{
12.     font-family: 'Roboto', sans-serif;
13. }
14.
15. header {
16.     width: 100%;
17.     height: 50px;
18.     background: #333;
19.     color: #fff;
20.
21.     position: fixed;
22.     top: 0;
23.     left: 0;
24.     z-index: 100;
25. }
26.
27. .contenedor {
28.     width: 98%;
```

```

29.         margin: auto;
30.     }
31.
32.     h1 {
33.         float: left;
34.     }
35.
36.     header .contenedor {
37.         display: table;
38.     }
39.
40.     form {
41.         width: 450px;
42.         margin: auto;
43.         background: rgba(0,0,0,0.4);
44.         padding: 10px 20px;
45.         box-sizing: border-box;
46.         margin-top: 20px;
47.         border-radius: 7px;
48.     }
49.
50.     table {
51.         width: 450px;
52.         margin: auto;
53.         background: rgba(0,0,0,0.4);
54.         padding: 10px 20px;
55.         box-sizing: border-box;
56.         margin-top: 20px;
57.         border-radius: 7px;
58.     }
59.
60.     input, textarea {
61.         width: 100%;
62.         margin-bottom: 20px;
63.         padding: 7px;
64.         box-sizing: border-box;
65.     }
66.
67.     #boton{
68.         background: #31384A;
69.         color: #fff;
70.         padding: 20px;
71.     }
72.
73.     #boton:hover{
74.         cursor: pointer;
75.     }
76.

```

## Menú.css

```
1. #menu-bar{
2.     display: none;
3. }
4.
5. header label {
6.     float: right;
7.     font-size: 28px;
8.     margin: 6px 0;
9.     cursor: pointer;
10.
11. }
12.
13. .menu{
14.     position: absolute;
15.     top: 50px;
16.     left: 0;
17.     width: 100% ;
18.     height: 100vh;
19.     background: rgba(51,51,51,0.9);
20.     transition: all 0.5s;
21.     transform: translateX(-100%);
22.
23. }
24.
25. .menu a{
26.     display: block;
27.     color: #fff;
28.     height: 50px;
29.     text-decoration: none;
30.     padding: 15px;
31.     border-bottom: 1px solid rgba(255,255,255,0.3)
32. }
33.
34. .menu a:hover{
35.     background: rgba(255,255,255,0.3);
36. }
37.
38. #menu-bar:checked ~ .menu{
39.     transform: translateX(0%);
40. }
41.
```

## BIBLIOGRAFIA

- Raspberry Pi. (n.d.). Consultat el 10 de Març de 2019 des de <https://www.raspberrypi.org/>
- Wikipedia - Arduino. (2019). Consultat el 10 de Març de 2019 des de <https://es.wikipedia.org/wiki/Arduino>
- Wikipedia - Raspberry Pi. (2019). Consultat el 10 de Març de 2019 des de [https://es.wikipedia.org/wiki/Raspberry\\_Pi](https://es.wikipedia.org/wiki/Raspberry_Pi)
- Prometec. (2019). Crear un servidor en la Raspberry Pi | Tienda y Tutoriales Arduino. Consultat el 15 de Març de 2019 des de <https://www.prometec.net/raspberry-pi-servidor/>
- Marc Meseguer. (n.d.). Servidor casero Raspberry Pi: Acceder al servidor desde el exterior. Consultat el 15 de Març de 2019 des de <https://sobrebts.com/montar-un-servidor-casero-con-raspberry-pi-parte-6-acceder-al-servidor-desde-el-exterior/>
- Fernán. (2015). Estación meteo | Tienda y Tutoriales Arduino. Consultat el 17 de Març de 2019, des de [https://www.prometec.net/proyecto\\_1/](https://www.prometec.net/proyecto_1/)
- Prometec. (2019). Subir datos a un servidor mediante WiFi | Tienda y Tutoriales Arduino. Consultat el 20 de Març de 2019 des de <https://www.prometec.net/esp8266-subir-valores/>
- J&G Proyectos Web. (2015). Diseñando una web responsive desde 0 (sin plugins ni frameworks), HTML y CSS - 1ra parte - YouTube. Consultat el 22 d'Abril de 2019 des de <https://www.youtube.com/watch?v=nOjxVmN-obU&t=1389s>
- Como proteger una página web con contraseña con PHP. (n.d.). Consultat el 15 de Maig de 2019 des de <https://norfipc.com/codigos/como-proteger-pagina-web-contrasena-php.php>
- Formularios básicos en HTML. (n.d.). Consultat el 17 de Maig de 2019 des de <http://www.htmlquick.com/es/tutorials/forms.html>
- Estilizando formularios HTML - Aprende sobre desarrollo web | MDN. (n.d.). Consultat el 22 de Maig de 2019 des de [https://developer.mozilla.org/es/docs/Learn/HTML/Forms/Styling\\_HTML\\_forms](https://developer.mozilla.org/es/docs/Learn/HTML/Forms/Styling_HTML_forms)